



A Developer's Guide to Using OpenTelemetry with ASP.NET Core for Enhanced Application Monitoring



A DEVELOPER'S GUIDE TO USING OPENTELEMETRY WITH ASP.NET CORE FOR ENHANCED APPLICATION MONITORING

STACKIFY

www.stackify.com

In today's competitive software landscape, providing a seamless user experience requires robust application monitoring. Developers rely on powerful tools and frameworks to gain insights into their applications' performance and detect anomalies. OpenTelemetry, an open-source observability framework, has emerged as a game-changer in this domain. When integrated with ASP.NET Core, OpenTelemetry can elevate your application's monitoring capabilities, enabling you to deliver exceptional performance and reliability.

In this guide, we'll explore how to use OpenTelemetry with ASP.NET Core, highlight its benefits, and discuss the importance of application log monitoring tools in a modern development ecosystem.

What is OpenTelemetry?

OpenTelemetry is an open-source observability framework designed to collect telemetry data—traces, metrics, and logs—from your applications. It standardizes the way developers instrument code and provides vendor-agnostic integrations with monitoring platforms.

Key features of OpenTelemetry include:

- **Cross-language support:** Compatible with multiple programming languages, including .NET.
- **Interoperability:** Works with various backends such as Jaeger, Prometheus, and Zipkin.
- **Extensibility:** Easily extendable to meet specific observability needs.

By using OpenTelemetry, developers can gain deep insights into their applications, streamline debugging, and optimize performance.

Why Use OpenTelemetry with ASP.NET Core?

ASP.NET Core is a powerful, cross-platform framework for building modern web applications. When combined with OpenTelemetry, it allows developers to:

- **Track end-to-end requests:** Monitor the flow of requests through different components.
- **Detect bottlenecks:** Identify performance issues in APIs, databases, and services.
- **Enhance user experience:** Proactively address performance issues to ensure a smooth experience.

Setting Up OpenTelemetry in ASP.NET Core

To get started, you need to configure your [ASP.NET Core application](#) to work with OpenTelemetry. Follow these steps:

1. Install Required NuGet Packages

Begin by installing the necessary NuGet packages:

Install-Package OpenTelemetry.Extensions.Hosting

Install-Package OpenTelemetry.Instrumentation.AspNetCore

Install-Package OpenTelemetry.Instrumentation.Http

Install-Package OpenTelemetry.Exporter.Console

2. Configure OpenTelemetry in **Program.cs**

Modify the **Program.cs** file to configure OpenTelemetry:

```
using OpenTelemetry.Resources;
```

```
using OpenTelemetry.Trace;
```

```
var builder = WebApplication.CreateBuilder(args);
```

```
builder.Services.AddOpenTelemetryTracing(tracerProviderBuilder =>
```

```
{
```

```
    tracerProviderBuilder
```

```
        .SetResourceBuilder(ResourceBuilder.CreateDefault().AddService("MyAspNetCoreApp"))
```

```
        .AddAspNetCoreInstrumentation()
```

```
        .AddHttpClientInstrumentation()
```

```
        .AddConsoleExporter();
```

```
});
```

```
var app = builder.Build();
```

```
app.UseRouting();
```

```
app.MapControllers();
```

```
app.Run();
```

3. Instrument Your Code

OpenTelemetry automatically instruments many components of ASP.NET Core applications, including HTTP requests and dependencies. For custom traces, you can use the [ActivitySource](#) API:

using System.Diagnostics;

```
var activitySource = new ActivitySource("MyAspNetCoreApp");
```

```
using (var activity = activitySource.StartActivity("CustomOperation"))
```

```
{  
    // Your custom logic here  
  
    activity?.SetTag("key", "value");  
}
```

Exporting Telemetry Data

OpenTelemetry supports multiple exporters to send telemetry data to different backends. Some popular options include:

- **Console Exporter:** Outputs telemetry data to the console for debugging.
- **Jaeger Exporter:** Sends traces to a Jaeger backend.
- **Prometheus Exporter:** Exposes metrics for Prometheus scraping.

To use a specific exporter, install the corresponding NuGet package and configure it in [Program.cs](#).

Benefits of Application Log Monitoring Tools

While OpenTelemetry provides robust tracing and metrics capabilities, combining it with [application log monitoring tools](#) can take your observability to the next level.

Log monitoring tools help developers:

- **Centralize logs:** Aggregate logs from different services and applications in one place.
- **Search and filter:** Quickly locate relevant logs using advanced search features.
- **Set alerts:** Get notified of anomalies or errors in real time.

Integrating log monitoring with OpenTelemetry creates a holistic view of your application's health, enabling you to correlate logs, traces, and metrics seamlessly.

Best Practices for Using OpenTelemetry with ASP.NET Core

To maximize the benefits of OpenTelemetry, follow these best practices:

1. Optimize Sampling

Sampling controls the amount of telemetry data collected. Use a sampling strategy that balances performance and observability needs. For example:

```
tracerProviderBuilder.SetSampler(new AlwaysOnSampler());
```

2. Use Semantic Conventions

Follow OpenTelemetry's semantic conventions to standardize trace and log metadata. This improves data consistency and simplifies integration with monitoring tools.

3. Secure Telemetry Data

Ensure that sensitive information is not included in telemetry data. Use filters to sanitize data before exporting it:

```
tracerProviderBuilder.AddProcessor(new SimpleProcessor(new  
SensitiveDataFilter()));
```

4. Monitor Performance

Telemetry collection introduces some overhead. Monitor the performance impact and optimize your instrumentation accordingly.

Conclusion

OpenTelemetry and ASP.NET Core together provide a powerful foundation for building observable applications. By leveraging their capabilities and integrating them with application log monitoring tools, developers can gain comprehensive insights into their applications, ensure reliability, and deliver exceptional user experiences.

Start your journey with OpenTelemetry today and transform the way you monitor and manage your ASP.NET Core applications.



Start your free trial today:- <https://stackify.com/free-trial/>